REMARKS

Claims 1-10 are currently pending in the application. By this amendment, claims 1 and 6 are amended and claims 11-18 are added for the Examiner's consideration. The foregoing separate sheets marked as "Listing of Claims" shows all the claims in the application, with an indication of the current status of each .

The Examiner has rejected claims 1-10 under 35 U.S.C. §102(a) as being anticipated by U.S. Patent No. 6,263,498 to Alcorn et al. ("Alcorn"). This ground of rejection is respectfully traversed for the reasons that follow. Alcorn contains some terminology used in the present invention, but fails to address or disclose the present invention. The Examiner has misapplied Alcorn, as will become evident from the following discussion of what Alcorn teaches.

Alcorn discloses a methodology for managing application modifications in a distributed data processing system. In the scheme described by Alcorn, an application program has a portion (the "first portion") placed on a server and another portion (the "second portion") placed on the client. This is a common arrangement for an "application centric environment" (as described in connection with Fig. 1B of the present invention), and is old in the art. It should be noted – as explained in the present application in connection with Fig. 1D (page 4, line 17, to page 5, line 4) – that the present invention uses an application server, upon which are built services (i.e. dubbed "server bullets") "which are then made available for completion of an application" (page 4, lines 21-22). Thus, in the present invention the server is not a location for a "portion" of an application program (i.e. where the application program is constructed as an entity and a portion thereof resides on a server), but rather the server is the location of pre-built services which can then be used by an application resident on a client computer. These services (e.g. in the J2EE implementation) have an API (Application Program Interface), which is how the services are used by an application.

In practical effect, this treats the server as an "application server", as a source of basic functionality "bullets", rather like an application treats an operating system as a source of functionality, where the functionality can be more efficiently provided to applications by a common operating system, in contrast to functionality coded separately for each application. Typically, as described in Alcorn (col 1, lines 30-32), the functionality provided by an operating system includes hardware control. In the present invention, the "server bullet" functionalities are created by "building and deploying the most commonly used functions and services directly on the application server" (page 4, lines 22-24), thereby eliminating "the need to scale an entire application" (page 4, lines 24-25).

Alcorn's contribution to the prior art is the notion of "bean dipping", which is a technology developed by IBM in the late 1990's. Using the "dipping" technology functionality can be added to beans on the server "portion" of an application without having to change the many clients that access the server (col 9, lines 37-38). The methodology described in Alcorn first creates a subclass (in an object oriented platform) having all the methods, events and properties of the original class, but providing "before" and "after" notifications (col 10, lines 16-19). This is called "morphing". Then the "before" and "after" notifications are used to modify the behavior of the subclass and therefore, because of the inheritance attributes of object oriented systems, modify the behavior of the original class (col 10, lines 23-25). This is called "dipping". The dips modify the runtime behavior of the beans to which they are attached, but do not provide new interfaces (col 10, lines 45-47). What was visible to the client before dipping does not change; dipping can add behavior, but without adding new properties, events, or methods to the bean (col 10, lines 53-56).

Bean morphing and dipping has nothing to do with the present invention. What the Examiner has observed in Alcorn is the use of certain terminology having parallels in the present invention. For example, Alcorn uses the term "reusable", which is also used in the present invention. However, in Alcorn "reusability" refers

to the inheritance characteristic of classes in object oriented systems (col 7, lines 56-63). In the present invention, "server bullets" are also reusable, but in the sense of being usable through adaptation by different applications within a specified enterprise environment (page 4, lines 4-16). Similarly, Alcorn discusses changes in "business rules" that may be applied through the "dipping" process (col 6, lines 52-59), which appear similar to the claimed "business process requirements" of the present invention, whose changes are incorporated by repeating the completion process (page 4, lines 14-16). However, as is clear from the above discussion, the significance of changes in bean behavior from "dipping" a different business rule is that <u>no change</u> is made to the interface viewed by the client, thus avoiding changes in the many clients accessing data on the server. By contrast, the significance of changes in "business process requirements" in the present invention is that these changes are relatively easily accommodated in revised code <u>in the application residing on the client</u> through use of "server bullets", whose use avoids "the need to scale an entire application" (page 4, lines 24-25). In the present invention, multiple applications may be efficiently scaled for revised business process requirements by recoding with reuseable "server bullets", whereas in Alcorn multiple clients of a single application can <u>avoid recoding of the client portion</u> by using bean "dipping" to change server bean behavior <u>without</u> changing the bean interface as viewed by the multiple clients.

As should by now be clear, the Alcorn reference is not aptly applied against the present invention. Alcorn neither discloses nor suggests the construction and use of "server bullets", i.e. reusable service components (page 9, lines 12-13) <u>built and deployed on the server to facilitate scaling</u> (page 4, lines 22-25). The claims have been amended to clarify the foregoing points and extended to more fully claim the invention.

It should further be noted that the Examiner's argument with respect to claims 5 and 10 is misplaced. The Examiner argues that the morphing tool "wrapping" of a bean in an API, i.e. the "before" and "after" notifications used for "dipping" (col 10,

Server Bullet technology   10/617,360   07980001aa
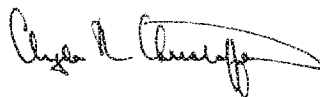Amendment dated 12/05/2006   Reply to office action mailed 06/05/2006

lines 15-24), is the same as "shrink wrapping". This is clearly not the case. As is
understood by those skilled in the art, and as used in the specification at page 3, line
19, page 5, line 9, and in connection with Fig. 3 at page 8, lines 10-13, "shrink
wrapping" refers to packaging of a software product in a plastic seal for distribution
in marketing channels. Claims 5 and 10 add the limitation of a shrink-wrap of
"server bullets" in the vertical pre-configuration. This has nothing to do with
Alcorn's morphing tool "wrapping".

In view of the foregoing, it is requested that the application be reconsidered,
that claims 1-18 be allowed, and that the application be passed to issue.

Should the Examiner find the application to be other than in condition for
allowance, the Examiner is requested to contact the undersigned at 703-787-9400
(fax: 703-787-7557; email: clyde@wcc-ip.com) to discuss any other changes deemed
necessary in a telephonic or personal interview. Please charge any deficiencies in
fees and credit any overpayment of fees to Attorney's Deposit Account No. 50-2041.

Sincerely,

Clyde R Christofferson
Reg. No. 34,138

Whitham, Curtis, Christofferson & Cook, P.C.   **Customer No. 30743**
11491 Sunset Hills Road, Suite 340
Reston, VA 20190
703-787-9400
703-787-7557 (fax)